Volume 22 (2003), Number 3

Adaptive Logarithmic Mapping For Displaying High Contrast Scenes

F. Drago, [1] K. Myszkowski, [2] T. Annen [2] and N. Chiba [1]

[1]Iwate University, Morioka, Japan. [2] MPI Informatik, Saarbrücken, Germany.

Abstract

We propose a fast, high quality tone mapping technique to display high contrast images on devices with limited dynamic range of luminance values. The method is based on logarithmic compression of luminance values, imitating the human response to light. A bias power function is introduced to adaptively vary logarithmic bases, resulting in good preservation of details and contrast. To improve contrast in dark areas, changes to the gamma correction procedure are proposed. Our adaptive logarithmic mapping technique is capable of producing perceptually tuned images with high dynamic content and works at interactive speed. We demonstrate a successful application of our tone mapping technique with a high dynamic range video player enabling to adjust optimal viewing conditions for any kind of display while taking into account user preference concerning brightness, contrast compression, and detail reproduction.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Image Processing and Computer Vision]: Image Representation

1. Introduction

Mainstream imaging and rendering software are now addressing the need to represent physically accurate lighting information in the form of high dynamic range (HDR) textures, environment maps, light fields, and images in order to capture accurate scene appearance. Clearly, proper capture of luminance (radiance) and chroma for any environment requires higher precision than offered by a 24-bit RGB representation. This fact has been early recognized by the lighting simulation[8] and physically based rendering[16, 20] communities. As a result of lighting computation, luminance values in the scene are reconstructed and rendered images are saved using file formats capable of representing the complete visible spectrum[19, 6, 5]. The same formats are used for high dynamic range imaging[2], where photographs of a static scene taken at different exposures are assembled and saved in a radiance map (Figure 1). Initially, HDR images have been used by Debevec[3] as a lighting tool to render CG objects illuminated in a real world setting. However, this format was soon adopted by photographers, who were finally able to cope with high contrast scenes. Modern digital cameras are moving toward greater contrast representation. Already consumer oriented cameras offer 12-bits or more data per channel and recent innovative chip design permits to achieve much more e.g., the Super CCD SR developed by Fuji, which incorporates both large, high-sensitivity S-pixels and smaller R-pixels for expanded dynamic range. Modern graphics acceleration cards also start to offer a HDR data representation using floating point precision throughout their rendering pipelines. We can envision that in a near future, the complete imaging pipeline will be based on physically accurate data.

Figure 1: *Dynamic range = 394,609:1. HDR image built from three stitched photographs taken at five different exposures.*

© The Eurographics Association and Blackwell Publishers 2003. Published by Blackwell Publishers, 108 Cowley Road, Oxford OX4 1JF, UK and 350 Main Street, Malden, MA 02148, USA.

Unfortunately, displaying methods have not progressed in a similar pace. Except for a few specialized devices, CRT and flat panel displays are still limited to a very small dynamic range, often less than 100:1, while the dynamic range of scenes represented by HDR images can span over five or more orders of magnitude.

Tone mapping is introduced in the graphic pipeline as the last step before image display to address the problem of incompatible luminance ranges. The question answered by most of the tone mapping algorithms developed for computer graphics applications is: *"Within the physical limitations of displaying hardware, how to present images perceptually similar to the original scenes to human viewers?"* Essentially, tone mapping should provide drastic contrast reduction from scene values to displayable ranges while preserving the image details essential to appreciate the scene content.

The tone mapping problem was first addressed by Tumblin and Rushmeier[16] and Ward[20]. They developed global mapping functions backed by results in psychophysics on brightness and contrast perception. Later, Ward[7] proposed the Histogram Adjustment technique which allocates dynamic range space in proportion to the percentage of pixels with similar brightness, again taking contrast perception into account. Some researchers focused simply on computation efficiency, mostly ignoring characteristics of the human visual system (HVS)[14]. Each of these methods can be classified as *spatially uniform* because a single mapping function is derived and used for all pixels in a given image. The tone mapping proposed in this paper belongs to this category.
Another group, the *spatially varying* methods, often attempt to model spatial adaptation by using locally changing mapping functions, which depend on a given pixel neighborhood. While spatially varying methods might produce the most compelling images, they are significantly more expensive than spatially uniform techniques and their use in interactive applications has not been shown so far. An interested reader can refer to a recent extensive survey on this topic[4].

Our motivation for this work is to address the need for a fast algorithm suitable for interactive applications which automatically produces realistically looking images for a wide variation of scenes exhibiting high dynamic range of luminance. For the sake of efficiency we use a spatially uniform tone mapping function which is based on a simple model of brightness perception. We provide the possibility of interactively tuning image appearance in terms of brightness and contrast while viewing an animation. The resulting images are detailed, and faithful representations of the original high contrast scenes reproduced within the capabilities of the displaying medium. The paper is organized as follows: Section 2 briefly describes the research results our technique is based upon. In Section 3 we present the tone mapping function, its parameters and usage. Section 4 proposes a solution to the loss of detail in dark areas caused by gamma correction. In Section 5 we discuss some essential optimizations leading to the implementation of a HDR movie player which enables the presentation of high dynamic range content in realtime. Finally, we conclude this paper and propose possible directions for future research.

2. Background

The term brightness B describes the response of the HVS to stimulus luminance L. This response has the form of compressive non-linearity which can be approximated by a logarithmic function (Weber-Fechner law) $B = k_1 \ln(L/L_0)$, where L_0 denotes the luminance of the background and k_1 is a constant factor. The relation has been derived in psychophysical threshold experiments through examining just noticeable differences ΔL for various L_0. Slightly different relations between B and L have been obtained depending on such factors as stimulus size, L_0, and temporal presentation. For example, supra-threshold experiments resulted in an observation that equal ratios of luminance lead to equal ratios of brightness and the HVS response should be rather modeled by a power function (Stevens law) $B = k_2 L^n$, where n falls in the range of 0.3 to 1.0. In practice, both descriptions are relatively close so that it is difficult to discriminate between them experimentally[18]. Therefore, we assumed the logarithmic relation in our tone mapping solution following Stockham[15] who recommended such a relation for image processing purposes:

$$L_d = \frac{\log(L_w + 1)}{\log(L_{max} + 1)} \tag{1}$$

where for each pixel, the displayed luminance L_d is derived from the ratio of world luminance L_w and maximum luminance in the scene L_{max}. This mapping ensures that whatever the dynamic range of the scene is, the maximum value is remapped to one (white) and other luminance values are smoothly incremented. While this formula leads to pleasant images, we found that the luminance compression is excessive and the feeling of high contrast content is lost.

3. Adaptive Logarithmic Mapping

The design of our tone mapping technique was guided by a few rules. It must provide consistent results despite the vast diversity of natural scenes and the possible radiance value inaccuracy found in HDR photographs. Additionally, it should be adaptable and extensible to address the current capabilities of displaying methods and their future evolution. Tone mapping must capture the physical appearance of the scene, while avoiding the introduction of artifacts such as contrast reversal or black halos. The overall brightness of the output image must be faithful to the context. It must be "user-friendly" i.e., automatic in most cases, with a few intuitive parameters which provide possibility for adjustments. It must be fast for interactive and realtime applications while avoiding any trade-off between speed and quality.

© The Eurographics Association and Blackwell Publishers 2003.

3.1. Scaling Scene Luminance to Image Brightness

The overall brightness of the output image is mainly determined by the lighting characteristics of the scene. It is then necessary to find an initial scalefactor from the scene luminance to output image brightness. We can make here an analogy with the photography where the exposure settings determine the appearance of the taken picture. Modern cameras offer different options for automatic exposure setting, such as center-weighted, center-spot, or matrix-metering. In the same fashion, we rely on two methods suitable for different use. For static images or when a user does not directly interact with the scene, we compute the logarithmic average of the scene based on luminance values for all pixels, similar to Tumblin and Rushmeier[16] (who called this scalefactor *world adaptation luminance*) or Reinhard et al.[12]. We also use a center-weighted scalefactor for interactive tone mapping or walkthrough sequences when a user's center of attention might shift from one location to another in the scene[13]. Our implementation of a center-weighted scalefactor calculates the logarithmic average of the region centered at a pixel (the center of viewing fixation) and convolved by a two-dimensional Gaussian distribution kernel. The area of the sampled region and Gaussian kernel is set by default to 15% of the scene area but can be adjusted interactively. This method might be used in conjunction with an eye tracking system. We also offer an exposure scale factor allowing users to adjust the brightness of the output image to their displaying conditions.

3.2. Contrast Adjustment

The principal characteristic of our tone mapping function is an adaptive adjustment of logarithmic base depending on each pixel's radiance. We interpolate luminance values found in the scene from $\log_2(L_w)$ to $\log_{10}(L_w)$. This essentially provides for good contrast and detail preservation in dark and medium areas while permitting maximum compression of high luminance values. In principle, a narrower or wider interval of logarithmic bases could be used, but we could not find any practical reason for it. The values of $\log_x(L_w)$ for $x < 2$ increase sharply making exposure adjustments difficult. On the other hand for $x > 10$ luminance compression is only marginally augmented and the overall image loses too much contrast. We also observed some color shift caused by high logarithmic bases.

Figure 2 shows the difference between images which are tone mapped with $\log_2()$ and $\log_{10}()$ functions after applying an initial world adaptation scalefactor. The following basic property of logarithm permits an arbitrary choice of logarithmic base:

$$\log_{base}(x) = \frac{\log(x)}{\log(base)} \tag{2}$$

For smooth interpolation among logarithmic bases, we rely upon Perlin and Hoffert "bias" power function[9]. Bias was first presented as a density modulation function, to change

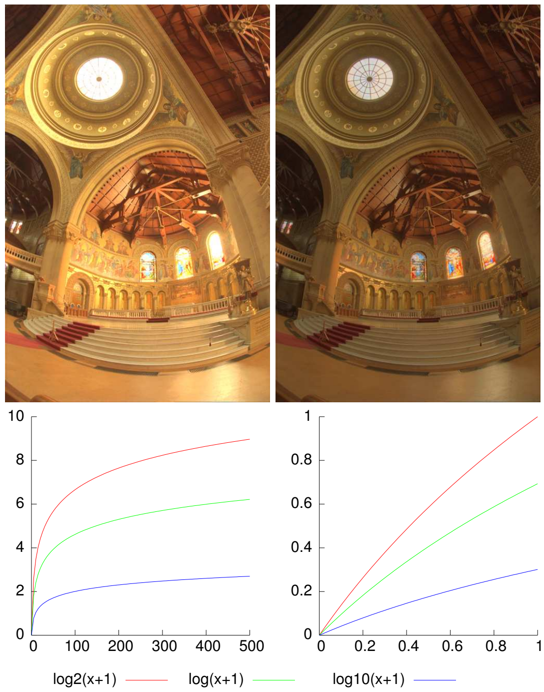

Figure 2: *The Stanford* MEMORIAL *church mapped with fixed base two logarithm (left) and with decimal logarithm (right). The contrast and brightness difference is evident but none of these images provides a satisfying rendition. Our tone mapping function offers the possibility to combine the characteristics of both images in a single result. Plots of the logarithm function show the difference among common logarithmic bases.* $\log_2()$ *increases sharply providing high contrast while* $\log_{10}()$ *drastically compresses higher values.*

density of the soft boundary between the inside and outside of a procedural hypertexture. It became a standard tool for texture synthesis and is also used for many different tasks throughout computer graphics. The bias function is a power function defined over the unit interval, an intuitive parameter b remaps an input value to a higher or lower value.

$$bias_b(t) = t^{\frac{\log(b)}{\log(0.5)}} \tag{3}$$

Figure 3 shows the bias curve for different b values. It is interesting to notice that $bias_{0.73}$ produces approximately the same mapping as the gamma correction function with a parameter $\gamma = 2.2$.

3.3. Algorithm

The input data is converted from its original format to a floating point representation of linear RGB values. Since the scene illuminant characteristic is not accurately known in most cases, we assume a D_{65} white point, to further convert tristimulus values between Rec.709 RGB and CIE XYZ.

The XYZ luminance component Y of each pixel (L_w for world luminance) and the maximum luminance of the scene

© The Eurographics Association and Blackwell Publishers 2003.

Figure 4: *The Stanford* MEMORIAL *Church processed with different bias parameters:* $b = 0.65$, $b = 0.75$, $b = 0.85$, *and* $b = 0.95$ *(from left to right). The scene dynamic range is 343,111:1. Radiance map courtesy of Paul Debevec.*

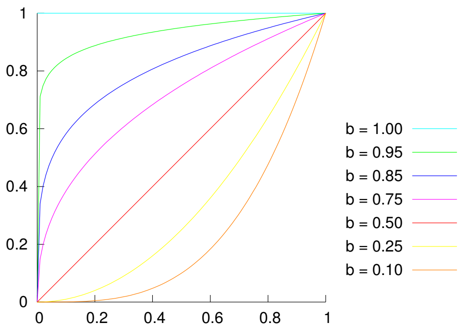

Figure 3: *The bias power function for different value of the parameter b (refer to Equation (3)). In our application, useful value for b falls into the range 0.5–1.0.*

L_{wmax} are divided by the world adaptation luminance L_{wa} and eventually multiplied by an exposure factor set by the user.

The tone mapping function presented in Equation (4) is used to compute a displaying value L_d for each pixel. This function is derived by inserting Equation (3) into the denominator of Equation (2). Equation (4) requires luminance values L_w and L_{wmax} (scaled by L_{wa} and the optional exposure factor) which characterize the scene as well as L_{dmax} which is the maximum luminance capability of the displaying medium. The parameter of the bias function is denoted by b (refer to Equation (3)).

$$L_d = \frac{L_{dmax} \cdot 0.01}{\log_{10}(L_{wmax}+1)} \cdot \frac{\log(L_w+1)}{\log\left(2+\left(\left(\frac{L_w}{L_{wmax}}\right)^{\frac{\log(b)}{\log(0.5)}}\right)\cdot 8\right)} \tag{4}$$

L_{dmax} is used as a scalefactor to adapt the output to its intended display. In the denominator decimal logarithm is used since the maximum luminance value in the scene is always re-sampled to decimal logarithm by the bias function. We use a value for $L_{dmax} = 100\,cd/m^2$, a common reference value for CRT displays. The bias parameter b is essential to adjust compression of high values and visibility of details in dark areas. The result of different values for parameter b are visible in Figures 4 and 6. The graph in Figure 5 shows the curves of the mapping function for a scene with maximum luminance of $230\,cd/m^2$.

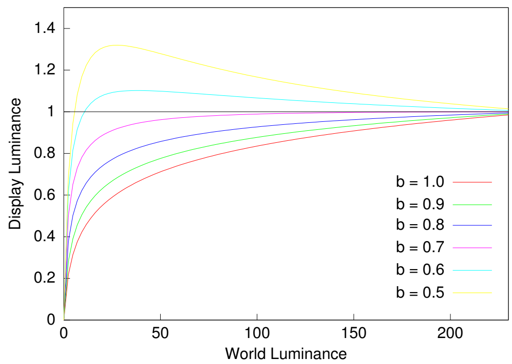

Figure 5: *Example plots of the tone mapping function (refer to Equation (4)) for* $L_{wmax} = 230\,cd/m^2$*. The maximum displayable value (white) is 1. For bias parameters smaller than* $b = 0.7$ *clamping to* L_{dmax} *will occur.*

Values for b between 0.7 and 0.9 are most useful to generate perceptually good images, but ideally a unique fixed parameter appropriate for most situations is needed. In an informal evaluation, we asked five persons to interactively choose from six different scenes, among four images tone mapped with different bias parameters (Figure 4 was a part of the survey) the ones they rated has looking the most realistic and the most pleasing. In terms of preference a bias parameter around 0.85 was consistently proposed. Results for realism were scattered for different images but consistent for each

© The Eurographics Association and Blackwell Publishers 2003.

subject. Averaging preferences and realism from this simple evaluation, we propose a default bias parameter $b = 0.85$. This indeed produces consistent, well balanced images with any kind of scenes. A side effect of changing the bias pa-

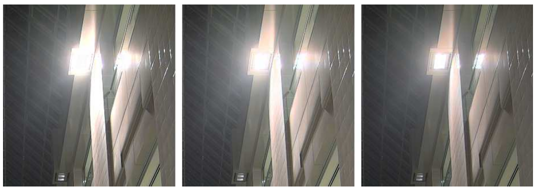

Figure 6: *Closeup of a light source of the* ATRIUM *scene at Aizu University (refer to Figure 8 for the full view of this scene in daylight conditions). This figure illustrates how high luminance values are clamped to the maximum displayable value. The images were computed using the following bias parameter values: $b = 0.5$, $b = 0.7$, and $b = 0.9$ (from left to right). The scene dynamic range is 11,751,307:1.*

rameter is some brightness fluctuation of the output image. The image brightness is approximately doubled for $b = 0.85$ and tripled for $b = 0.7$ in respect to images for $b = 1.0$. This affects the realism of images even though the increase of contrast due to the bias value reduction naturally leads to brighter images. We introduce here a scalefactor to world adaptation luminance, aiming at keeping a constant brightness impression. Again the adaptation is based on a default bias parameter equal to 0.85:

$$L_{wa} = L_{wa}/(1 + b - 0.85)^5$$

Figures 4 and 6 benefited from this scalefactor, the global brightness impression is almost constant, even though the contrast among images is very different.

4. Gamma Correction

Gamma correction must be applied to the tone mapped data to compensate for the non-linearity of displaying devices. A gamma coefficient $\gamma = 2.2$ is usually used for non-corrected displays, the gamma correction function is $L_d = L_w^{1/\gamma}$. In our pipeline, this correction is applied to linear RGB tristimulus values after tone mapping and conversion from CIE XYZ. We would like to address a potential problem of the gamma function. At the origin's vicinity, the gamma function exhibits a very steep slope. Even though we use floating point precision [1], after correction and quantization to 24 bit, originally dark pixels will all be transformed to medium values. This results in significant contrast and detail loss in shadowed areas. Ward's Histogram Adjustment method[7] ensures that all displayable values are represented in the final image. However, other tone mapping methods potentially suffer from this phenomenon and the contrast of their resulting images might be improved by considering this problem. Gamma functions with better perceptual accuracy have been

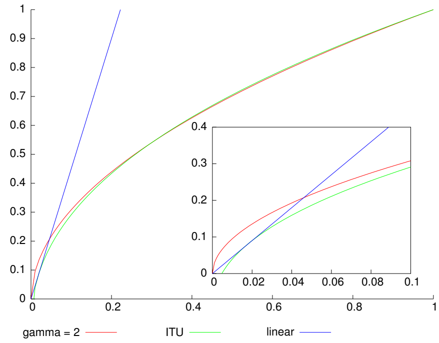

Figure 7: *Comparison of the gamma power function usually used in computer graphics with the ITU-R BT.709 transfer function. The essential difference is the less drastic mapping of dark pixels in the ITU-R BT.709 function.*

proposed, for example the sRGB color space includes a specific transformation. The international standard recommendation is the ITU-R BT.709 transfer function[11]; it describes the transformation done by a video camera to produce the best possible images on a calibrated display. This function is close to $\gamma = 2$, but assumes a $\gamma = 1.125$ correction of the display to compensate for the viewing environment. The principal differences between the gamma power function and the ITU-R BT.709 transfer function (refer to Figure 7) are smaller output values for dark pixels in the latter case. This results in better contrast and details in dark areas and potential attenuation of the noise often found in dark parts of photographs.

In its original form, the ITU-R BT.709 gamma correction is:

$$E' = \begin{cases} 4.5L & L \le 0.018 \\ 1.099L^{0.45} - 0.099 & L > 0.018 \end{cases}$$

where L is the linear value of each RGB tristimulus and E' the nonlinear pixel value to be displayed.

The ITU-R BT.709 transfer function has fixed parameters. This lacks convenience for computer graphics applications, where different transfer values might be needed depending on the lighting conditions surrounding the display, custom user settings, and operating system. We adapted the function to use familiar γ values and we use a simple fit of the linear segment at the origin. Our transfer function based on the ITU-R BT.709 standard is:

$$E' = \begin{cases} slope \cdot L & L \le start \\ 1.099L^{\frac{0.9}{\gamma}} - 0.099 & L > start \end{cases}$$

Where $slope$ is the elevation ratio of the line passing by the origin and tangent to the curve, and $start$ is the abscissa at the point of tangency. All the images in this paper have been corrected with this custom gamma function using $\gamma = 2.2$. Direct comparison of images shows some contrast enhancement in dark to medium areas, while keeping a similar overall brightness.

© The Eurographics Association and Blackwell Publishers 2003.

5. Implementation and Results

In our original scheme, each pixel is processed with Equation (4), and computation time increases linearly with the number of pixels. This is too slow for realtime applications so we have been looking for a faster solution. We rejected the idea of using a lookup table because it would need too many entries to satisfy the wide dynamic range, and a different table is needed if the user changes a parameter in the tone mapping procedure.

Evaluation of the bias function for each pixel is particularly expensive. We found that if the luminance difference among pixels is below a certain threshold, the bias function can be evaluated just once for their average luminance value. In practice, we split the input image into 3×3 pixel tiles and perform the bias computation for each group of nine pixels. This adaptation gives a substantial speedup even for very detailed scenes (e.g., MEMORIAL) and performs even better for simple scenes such as the computer generated ROOM.

Further acceleration was obtained by the Pade approximation of $\log(x+1)$ for low radiance values. Table 1 shows the computation time of our algorithm for five scene (refer to Figures 4 and 8) of different sizes, the speedup factor resulting from our optimizations, and the percentage of error introduced by those optimizations. In all the images tested, we could not visually detect a difference between the results of the original algorithm and the faster version.

Scene	Size (pixels)	Base (sec.)	Fast (sec.)	Speedup # times	Diff (%)
MEMORIAL	512×768	0.143	0.036	3.97	0.75
NAVE	720×480	0.126	0.031	4.06	0.21
ROOM	3000×1950	2.111	0.428	4.93	0.43
ATRIUM	1016×760	0.281	0.061	4.61	0.22
PANORAMA	2000×901	0.652	0.153	4.26	0.14

Table 1: *Tone mapping routine execution time before and after optimization, speed increase, and the RMS image difference. Algorithm running on a Pentium IV 2.2GHz, compiled with Intel C compiler version 5. The tone mapping routine is taken into account here. Image IO, color space transformation, and the initial calculation of L_{wa} remain constant and are not part of the table.*

5.1. A High Dynamic Range Movie Player

In some multimedia applications, it can be more convenient to distribute video streams in an HDR format. This way, end-users are able to tune display parameters to accommodate for their hardware characteristics as well as external lighting conditions. Also, the animation might be adjusted according to personal preference, achieving a desired balance between reproduced contrast, details, and brightness. At present, these settings must be decided by the distributor and are fixed for streamed video.

To address those issues we implemented our tone mapping method in a HDR movie player. Instead of saving already tone mapped 24 bit images, we created a HDR movie file format in which the physical radiance value of each pixel is available. This offers the same advantages and capabilities as for static HDR scenes. The viewer has access to a perceptually tuned rendition of the movie through tone mapping while conserving the original HDR data. Beside exposure adjustment, setting the maximum display luminance L_{max} makes optimal viewing possible with any kind of display, and interactive adjustment of the bias parameter results in different luminance and contrast compression.

We use Ward's RGBE format[19] to build the movie file. Instead of using three floating point values (96-bits), each pixel is represented by four integers (32-bits). The obvious advantage is a 2/3 file size reduction, allowing to save long animations and reading frames from disk in a manageable time. The downside is the exponentiation needed to convert data for tone mapping and display. Also, a major speed bottleneck is the necessary conversion from RGB tristimulus to luminance values and back. For each frame, we precompute and save constants needed for tone mapping, such as world adaptation L_{wa} and maximum luminance L_{wmax}. These of course would have to be calculated while playing for a VRML application or an animation rendered in realtime. We simulate the time-dependent adaptation of the human visual system by a weighted averaging of the world adaptation of the last four frames with the current.

We implemented two versions of the tone mapping function. A straightforward C routine, and a GPU implementation using the OpenGL fragment program extension. Modern GPUs support SIMD (Single Instruction Multiple Data) instruction sets capable of performing very fast parallelized floating point calculations. The ATI Fire GL X1 which we used has eight pixel pipeline and a 256 bit memory interface, which means that eight pixels are processed in parallel making the GPU very powerful even at a much slower clock rate than modern CPUs. Table 2 summarizes the frame rates obtained on our test PC.

Resolution	Software fps	Hardware fps
640 × 480	12.3	23.5
320 × 240	41.2	74.3

Table 2: *Statistics for software and hardware implementations of our tone mapping operator. Values denote the average number of frames per second obtained while playing Paul Debevec's "Rendering with natural light" animation.*

6. Conclusions

We presented a perception-motivated tone mapping algorithm for interactive display of high contrast scenes. In our algorithm the scene luminance values are compressed using logarithmic functions, which are computed using different bases depending on scene content. The $\log_2$ function is used in darkest areas to ensure good contrast and visibility, while

© The Eurographics Association and Blackwell Publishers 2003.

the $\log_{10}$ function is used for the highest luminance values to reinforce the contrast compression. In-between, luminance is remapped using logarithmic values based on the shape of a chosen bias function. This scheme enables fast non-linear tone mapping without objectionable image artifacts. Although our technique is fully automatic, the user can interactively choose varying image details versus contrast by effectively changing the shape of the bias curve using an intuitive parameter. Because of the computation performance our tone mapping can be used for playing HDR video sequences while providing the user unprecedented control over the video brightness, contrast compression, and detail reproduction.

As future work we intend to perform perceptual experiments to determine an automatic bias value as a function of the scene content and its dynamic range of luminance. Our approach might be further extended by using different functions to interpolate between logarithmic bases. Also, the HVS models of temporal adaptation could be incorporated to our tone mapping algorithm to make the displaying of video sequences more realistic.

Acknowledgements

We would like to thank Paul Bourke for permitting us to use his OpenGL based stereo animation viewing software as a framework for our HDR movie player. Also, we would like to thank Paul Debevec, Greg Ward, Raanan Fattal, Jack Tumblin, and Gregory Downing for making their HDR images and animations available. We thank Grzegorz Krawczyk for help in measuring timings for the GPU version of our HDR video player. This work was supported partly by Telecommunications Advancement Organization of Japan within the framework of "A Support System for Region-specific R&D Activities" and by the European Community within the scope of the RealReflect project IST-2001-34744 "Realtime visualization of complex reflectance behavior in virtual prototyping".

References

1. J. Blinn. Dirty Pixels. *IEEE Computer Graphics & Applications*, **9**(4):100–105, 1989. 5
2. P.E. Debevec and J. Malik. Recovering High Dynamic Range Radiance Maps from Photographs. *Proceedings of ACM SIGGRAPH 97, ACM, 369–378, 1997.* 1
3. P.E. Debevec. Rendering Synthetic Objects Into Real Scenes: Bridging Traditional and Image-Based Graphics With Global Illumination and High Dynamic Range Photography. *Proceedings of ACM SIGGRAPH 98, ACM, 189–198, 1998.* 1
4. K. Devlin, A. Chalmers, A. Wilkie, and W. Purgathofer. Tone Reproduction and Physically Based Spectral Rendering. *Eurographics 2002: State of the Art Reports*, Eurographics, 101–123, 2002. 2
5. Industrial Light & Magic, OpenEXR, High Dynamic Range Image File Format. Lucas Digital Ltd, 2003 1
6. G.W. Larson. LogLuv Encoding for Full-Gamut, High-Dynamic Range Images. *Journal of Graphics Tools*. A K Peters Ltd., **3**(1):815–830, 1998. 1
7. G.W. Larson, H.E. Rushmeier, and C. Piatko. A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. *IEEE Transactions on Visualization and Computer Graphics*, **3**(4):291–306, 1997. 2, 5
8. N.J. Miller, P.Y. Ngai, and D.D. Miller. The Application of Computer Graphics in Lighting Design. *Journal of the Illuminating Engineering Society*, **14**(1):6–26, 1984 1
9. K. Perlin and E.M. Hoffert. Hypertexture. *Computer Graphics (Proceedings of ACM SIGGRAPH 89)*, ACM, **23**, 253–262, 1989. 3
10. C.A. Poynton. A Technical Introduction to Digital Video. John Wiley & Sons, 1996
11. Recommendation ITU-R BT.709-4, Parameter Values for the HDTV Standards for Production and International Programme Exchange, ITU, 2000 5
12. E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic Tone Reproduction for Digital Images, *ACM Transactions on Graphics*, **21**(3):267–276, 2002. 3
13. A. Scheel, M. Stamminger, and H-P. Seidel. Tone Reproduction for Interactive Walkthroughs. *Computer Graphics Forum*, **19**(3):301–312, 2000. 3
14. C. Schlick. Quantization Techniques for the Visualization of High Dynamic Range Pictures. *Photorealistic Rendering Techniques*, Springer-Verlag, 7–20, 1994. 2
15. T.G. Stockham. Image Processing in the Context of a Visual Model, *Proceedings of the IEEE*, **60**:828–842 2
16. J. Tumblin and H.E. Rushmeier. Tone Reproduction for Realistic Images. *IEEE Computer Graphics and Applications*, **13**(6):42–48, 1993. 1, 2, 3
17. J. Tumblin, J.K. Hodgins, and B.K. Guenter. Two Methods for Display of High Contrast Images. *ACM Transactions on Graphics*, **18**(1):56–94, 1999.
18. W.A. Wagenaar. Stevens vs. Fechner: a Plea for Dismisal the Case. *Acta Psychologica, Amst*, **39**, 225–235, 1975. 2
19. G. Ward. Real Pixels. *Graphics Gems II*. Academic Press, 80–83, 1991. 1, 6
20. G. Ward. A Contrast-Based Scalefactor for Luminance Display. *Graphics Gems IV*. Academic Press, 415–421, 1994. 1, 2

© The Eurographics Association and Blackwell Publishers 2003.

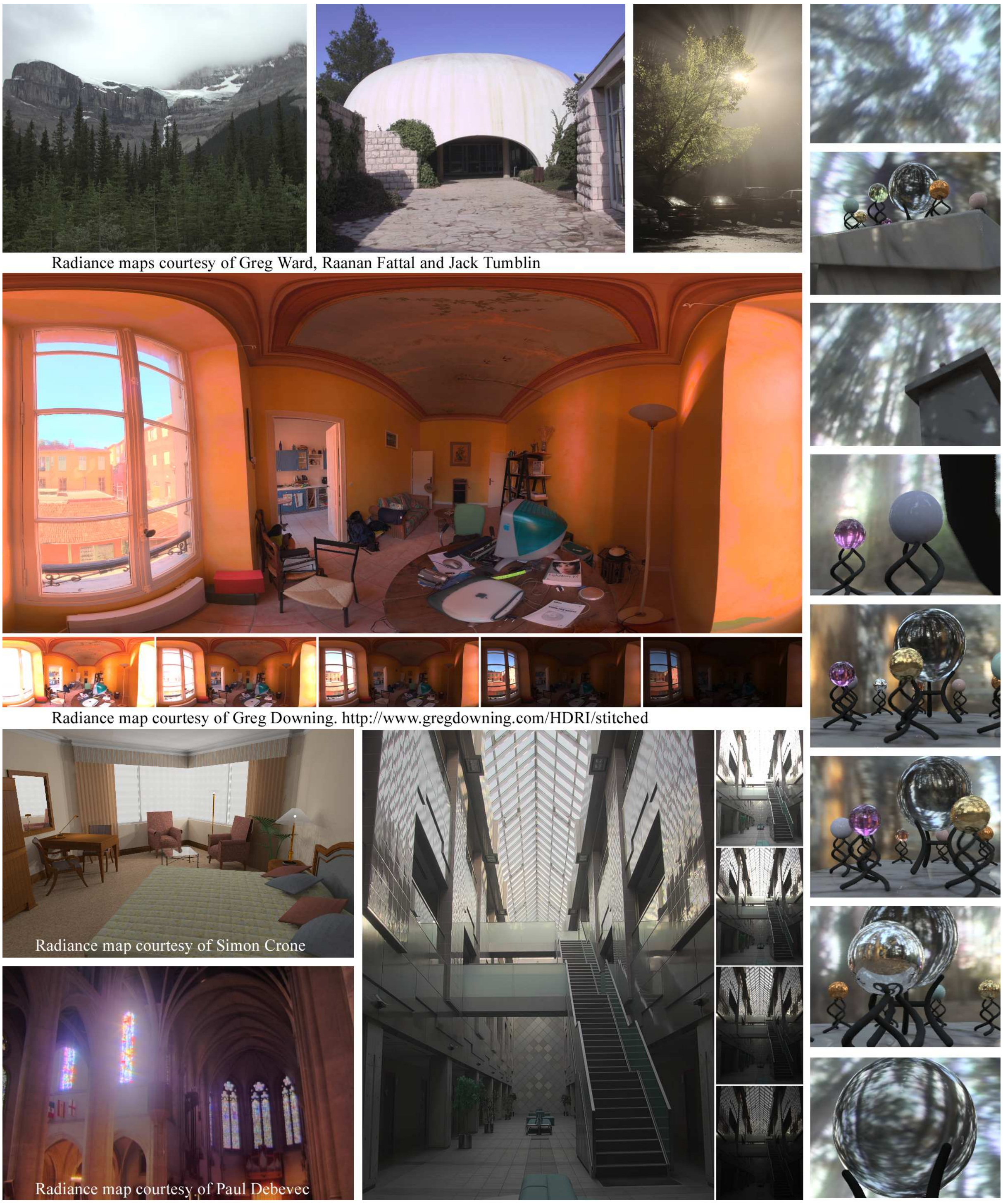

Figure 8: *A selection of high dynamic range images processed with our tone mapping function. Table 1 shows some statistics concerning the tone mapping computation for these images. On the right side, some frames captured from our HDR movie player showing Paul Debevec's "Rendering with natural light" animation.*

© The Eurographics Association and Blackwell Publishers 2003.